Enhancing Cybersecurity Through Automated Vulnerability Inspection

An In-Depth Analysis of Our Automated Application

Sai Pavani Chalamala  
 Computer Science  
 Southeast Missouri State University   
 Cape Girardeau MO US   
schalamala1s@semo.edu

Prasad Dama   
 Computer Science  
 Southeast Missouri State University   
 Cape Girardeau MO US  
pdama1s@semo.edu

Saurav Ghosh  
 Computer Science  
 Southeast Missouri State University   
 Cape Girardeau MO US  
 sghosh3s@semo.edu

Lindsey Redington  
 Computer Science  
 Southeast Missouri State University   
 Cape Girardeau MO US lmredington1s@semo.edu

ABSTRACT

As companies move toward remote employment, ensuring the security of the networks becomes significantly more complex and challenging. This complexity is primarily due to the variety of platforms that are integrated and utilized. Each point of access exposes the system to an entirely new set of security risks and in turn requires preventative measures. In these complex networks, identifying where to start implementing additional security measures can be overwhelming. With Automated Vulnerability Inspection cybersecurity professionals, specifically those responsible for implementing secure practices within the network, are able to obtain general information about vulnerabilities within their system. This inspection utilizes NMAP to identify CVEs within a network and then display the findings in a simplified and more readable format.

∗Article Title Footnote needs to be captured as Title Note

†Author Footnote to be captured as Author Note

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

*WOODSTOCK’18, June, 2018, El Paso, Texas USA*

© 2018 Copyright held by the owner/author(s). 978-1-4503-0000-0/18/06...$15.00

https://doi.org/10.1145/1234567890

CCS CONCEPTS

• Security Testing and Verification • Network Security • Threat Intelligence • Software Security Engineering • Vulnerability Management

KEYWORDS

Cybersecurity, Vulnerability scanning, Security assessment, Security tools, Security best practices

ACM Reference format:

Prasad Dama, Saurav Ghosh, Lindsey Redington, and Sai Pavani Chalamala. 2023. Enhancing Cybersecurity Through Automated Vulnerability Inspection: An In-Depth Analysis of Our Automated Application. *4 pages.*

1 Literature Review

The identification of the Common Vulnerabilities and Exposures (CVEs) is vital in protecting the confidentiality, availability, and integrity of applications. Consumers and producers are able to view CVEs in order to identify issues that need to be addressed, whether than is developing an update, downloading an existing update or version, or replacing the entire application. Rather than testing each system individually and independently finding vulnerabilities, the public disclosure allows CSOs (Chief Security Officers), or other parties responsible for security, to skip the taxing part of penetration testing.

Software included in a CVE:

1. Tooltalk database server
2. Internet Explorer 3.x and 4.x
3. Perl 4.x and 5.x
4. Apache httpd versions 1.1.1 and earlier
5. Netscape 2.x, 3.x and 4.x
6. Sendmail versions 8.8.3 and 8.8.4
7. FLEXlm LicenseManager versions 4.0 to 5.0
8. FTP server 3.5 [CVE-1999-0079]
9. NFS server [CVE-1999-0084]
10. Apache 1.2.5 and earlier [CVE-1999-0107]
11. Solaris 2.5.1 [CVE-1999-0109]
12. Adaptive Website Framework (AWF) 2.10 and earlier [CVE-2005-4372, CVE-2005-4373]
13. Mercury CMS 4.0 and earlier [CVE-2005-4406, CVE-2005-4407]
14. phpMyAdmin 2.7.0 [CVE-2005-4450]
15. Oracle Application Server (OracleAS) [CVE-2005-4549, CVE-2005-4550]
16. Debian Linux 2.1 [CVE-1999-1496]
17. POP3 Mail Server 2.3.1 [CVE-1999-1500]
18. SunOS 4.1 through 4.1.3 [CVE-1999-1507]
19. Linux kernel 2.2.x before 2.2.25 and 2.4.x before 2.4.21 [CVE-2003-0127]

Many businesses must create their own team of vulnerability management experts, or contract one, in order to consider the numerous CVEs relevant to their particular network/devices and achieve sufficient security (and consider the numerous relevant CVEs). A further step towards efficient testing and threat identification is provided by [3] which proposes a

“similarity measurement to compare and categorize vulnerabilities, and a set of security metrics to rank attacks based on vulnerability analysis.”

In addition to CVEs, the Cybersecurity Performance Goals (CPG) checklist provided by Cybersecurity and Infrastructure Security Agency (CISA) helps professionals develop a plan and methodology to ensure sufficient security of their network. CISA splits the goals into five categories:

* Identify
* Protect
* Detect
* Respond
* Recover

In this checklist, under each category, there are different goals. Under each goal the outcome, recommended action, TTP/risks addressed, scope, and other identifying categorizations like the cybersecurity framework (CSF) reference numbers from the National Institute of Standards and Technology (NIST).

2 Problem

When implementing and maintaining good cybersecurity practices it is difficult to determine where to start. As mentioned above, one helpful tool provided by the Cybersecurity and Infrastructure Security Agency (CISA) is the Cybersecurity Performance Goals (CPG) checklist. This outlines goals and standards that organizations should implement and achieve. Although this list points cybersecurity engineers and CSOs in the direction of the end result, it is difficult to figure out where the vulnerabilities are in the network or system. As companies move toward remote employment, ensuring the security of the networks becomes significantly more complex and challenging. This complexity is primarily due to the variety of platforms that are integrated and utilized. Rather than accessing the organization’s network locally, remote employees are accessing the same network from several different locations. Each point of access exposes the system to an entirely new set of security risks and in turn preventative measures. In these complex networks, identifying where to start implementing additional security measures can be overwhelming.

The other challenging factor for integrating the CPGs is how to solve or address the vulnerability. Cybersecurity professionals, specifically those responsible for implementing secure practices within the network, are able to obtain general information about vulnerabilities within their system through Automated Vulnerability Inspection. This inspection also includes references to the relevant CVEs of the discovered vulnerabilities. Linking CVEs to the system will also help an engineer to develop or implement a solution if one has been found. With this information they can develop a realistic view on the general status of their cybersecurity implementation within the network.

3 Design of Experiment

In order to first validate the functionality of the AVI application, a network was created which hosted a virtual machine built with a variety of vulnerabilities. When using the AVI application against the virtual machine, the vulnerabilities along with description, location and their corresponding CVEs and other details.

Given the information found from this, it was able to identify an out of date SQL Server which is vulnerable. Based on the CVE, and the information provided, the user would be able to conclude that a simple update could mitigate the vulnerability.

4 Implementation

4.1 Platform

For testing purposes, our program works for Kali and Kali-Linux operating systems. Executing the Java code was tested and successful on both Kali-Linux and Windows operating systems. The nmap scan is successful for any devices that support NMAP. However, some devices may have certain security measures in place that prevent the scan from gathering all of the available information and successfully complete the scan.

Tested Operating Systems:

* Windows 11
* Debian Linux

4.2 Language

1. Bash
2. Java (with JFlat library)

4.3 Tools

1. NMAP
2. Java Eclipse

5 Outline for Final Model

In the final model, given an IP address, the application first completes a nmap scan. In this scan we also retrieve information about vulnerabilities. Our application saves the scan into an xml file. This file includes:

* Ports
* Protocols
* Vulnerabilities (description)
* CVEs
* Results of vulnerability
* References
* Date discovered
* BIDs
* State
* Other additional information.

In the scan, information such as all of the possible SQL Injection Queries and CSRF vulnerabilities. All of this information can be overwhelming to read all at once. For this purpose, the application converts this XML into a simplified CSV file.

A diagram of a program

Description automatically generated

ACKNOWLEDGMENTS

Southeast Missouri State University - We appreciate your support of the research aspects of this project, providing a fertile ground for innovation and exploration.

The open-source communities whose software and tools underpin various components of the AVI Device, have been a cornerstone of this project's success. We thank the countless developers and contributors who maintain these critical resources.

REFERENCES

[1] “CPG: Cross-sector Cybersecurity Performance Goals – Performance Goals version: 1.0.1,” Cybersecurity and Infrastructure Security Agency, https://www.cisa.gov/sites/default/files/2023-03/CISA\_CPG\_REPORT\_v1.0.1\_FINAL.pdf (accessed Oct. 6, 2023).

[2] A. Patrizio, "Major security vulnerability found in top servers: More than 20 vulnerabilities have been found affecting unified extensible firmware interfaces (UEFI) software, allowing attackers to bypass hardware security mechanisms," *Network World* (Online), 2022. Available: https://ezproxy.semo.edu:2443/login?url=https://www.proquest.com/trade-journals/major-security-vulnerability-found-top-servers/docview/2626967799/se-2.

[3] J. A. Wang et al, "Measuring and ranking attacks based on vulnerability analysis," Information Systems and eBusiness Management, vol. 10, (4), pp. 455-490, 2012. Available: https://ezproxy.semo.edu:2443/login?url=https://www.proquest.com/scholarly-journals/measuring-ranking-attacks-based-on-vulnerability/docview/1197065399/se-2. DOI: https://doi.org/10.1007/s10257-011-0173-5.

Conference Name:ACM Woodstock conference

Contributions

Lindsey Redington:

* Wrote up document
* Participated in developing ideas for the document
* Helped develop code
* Demonstrated part of the application

Prasad Dama

* Created the diagram
* Participated in developing ideas for the document
* Helped develop code
* Demonstrated part of the application

Saurav Ghosh

* Participated in developing ideas for the document
* Developed code

Sai Pavani Chalamala

* Participated in developing ideas for the document

Conference Short Name:WOODSTOCK’18

Conference Location:El Paso, Texas USA

ISBN:978-1-4503-0000-0/18/06

Year:2018

Date:June

Copyright Year:2018

Copyright Statement:rightsretained

DOI:10.1145/1234567890

RRH: F. Surname et al.

Price:$15.00